

Webserver

in a nutshell

Sebastian Hollizeck

June, the 4th 2013

Definition of Webserver

HTTP

Just another definition

So what is it now?

Example

Static is no fun

CGI

php

comparison

log-file

Software

SSL

Webserver

a formal definition

A WEB SERVER IS A PROGRAM THAT, USING THE CLIENT/SERVER MODEL AND THE WORLD WIDE WEB'S HYPERTEXT TRANSFER PROTOCOL (HTTP), SERVES THE FILES THAT FORM WEB PAGES TO WEB USERS (WHOSE COMPUTERS CONTAIN HTTP CLIENTS THAT FORWARD THEIR REQUESTS).

HTTP?

What is that anyways?

THE HYPERTEXT TRANSFER PROTOCOL (HTTP) IS A
STATELESS APPLICATION PROTOCOL FOR DISTRIBUTED,
COLLABORATIVE, HYPERMEDIA INFORMATION SYSTEMS

HTTP!

it transfers!

HTTP/1.0 implements three methods:

GET retrieve the specified data from the server

POST send data to the server

HEAD like GET but just the header

HTTP!

and other stuff

HTTP/1.1 implements five more methods:

PUT save data on the server

DELETE delete data on the server

TRACE send the received data back

OPTIONS show possible methods for this resource

CONNECT form a TCP/IP tunnel

PATCH small changes to a resource

HTTP!

the stateless thing

Server does not need to store any information. All parameters needed are presented in the call.

Ways around:

- Hidden variables
- Cookies
- Query strings

HTTP!

an easy example

```
$ curl -v -I seba.tbl
* About to connect() to seba.tbl port 80 (#0)
*   Trying 192.168.16.12...
* connected
* Connected to seba.tbl (192.168.16.12) port 80 (#0)
> HEAD / HTTP/1.1
> User-Agent: curl/7.26.0
> Host: seba.tbl
> Accept: */*
>
```


HTTP!

an easy example cont'd

```
* HTTP 1.1 or later with persistent connection
< HTTP/1.1 200 OK
< Date: Sat, 01 Jun 2013 15:46:13 GMT
< Server: Apache/2.2.22 (Debian)
< Last-Modified: Sat, 01 Jun 2013 11:30:10 GMT
< ETag: "6008b-f31-4de160ddbd480"
< Accept-Ranges: bytes
< Content-Length: 3889
< Vary: Accept-Encoding
< Content-Type: text/html
< X-Pad: avoid browser bug
* Connection #0 to host seba.tbl left intact
* Closing connection #0
```

Webserver

non static

Displaying and storing data is not everything a web server can do.

CGI Common-Gateway-Interface

php PHP: Hypertext Preprocessor

CGI

is it common?

- connection between software and resources
- mostly realized via scripting languages
- easy usage

```
#!/bin/bash  
echo "Content-type: text/html"  
echo  
echo  
echo "Hallo, world!"
```

php

Is it better?

- easy database usage
- good http access
- high functionality

```
<?php  
    echo 'Hallo, world!';  
?>
```

php or CGI

Which is it?

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Hallo</title>
  </head>
  <body>
    <?php
      echo 'Hallo, world!';
    ?>
  </body>
</html>
```

```
#!/bin/bash
printf "Content-type: text/html\n\n"
echo "<html>"
echo "<head>"
echo "<title>Hallo</title>"
echo "</head>"
echo "<body>"
echo "Hallo, world!"
echo "</body>"
echo "</html>"
```

/var/log/apache2/errorlog

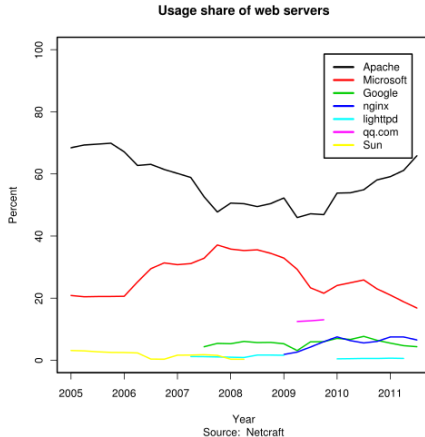
tailf is the method

When working with your web server always sneak a peak to the apache errolog!

THANK YOU FOR YOUR ATTENTION

Webserver

software and all



SSL

in a nutshell



Figure: schematic view of ssl